UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/819,772	03/28/2001	Michael Petrov	02509/90	2624

26646 7590 04/15/2009

KENYON & KENYON LLP
ONE BROADWAY
NEW YORK, NY 10004

EXAMINER	
CUNNINGHAM, GREGORY F	
ART UNIT	PAPER NUMBER
2624	
MAIL DATE	DELIVERY MODE
04/15/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte MICHAEL PETROV, ALEXANDER MIGDAL,
ALEXEI LEBEDEV, VERONICA SHELEKHOVA,
LEONID POLONSKIY, and VADIM ABADJEV

Appeal 2009-1413
Application 09/819,772
Technology Center 2600

Decided:[1] April 15, 2009

Before ROBERT E. NAPPI, JOHN A. JEFFERY, and
ELENI MANTIS MERCADER, *Administrative Patent Judges*.

JEFFERY, *Administrative Patent Judge*.

DECISION ON APPEAL

---

[1] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, begins to run from the decided date shown on this page of the decision. The time period does not run from the Mail Date (paper delivery) or Notification Date (electronic delivery).

Appellants appeal under 35 U.S.C. § 134 from the Examiner's rejection of claims 2-10, 55-63, and 114-117. We have jurisdiction under 35 U.S.C. § 6(b). We affirm-in-part.

STATEMENT OF THE CASE

Appellants invented a system and a method for restoring a previous version of a three-dimensional mesh model on a computer system. That is, the method enables a user to undo previously-performed operations by enabling the user to edit a working copy of the model using an ordered list of operations. The ordered list provides a complete record of every operation performed on the initial copy of the mesh to arrive at the current working copy. This technique reduces the amount of memory needed to maintain a record of prior versions of the mesh as editing progresses.[2] Claim 5 is illustrative:

5. A method for restoring a previous version of a three dimensional mesh model on a computer system comprising:

retrieving a stored copy of an earlier state of the three dimensional mesh model on the computer system;

retrieving an ordered list of operations on the computer system; and

performing at least some of the operations in the ordered list of operations on the retrieved copy of the three dimensional mesh model;

wherein the ordered list of operations contains the operations which if performed in order on the earlier state of the three dimensional mesh model would result in a current state of the three dimensional mesh model.

[2] *See generally* Spec. 8.

The Examiner relies on the following prior art reference to show unpatentability:

Kermit Sigmon, *MATLAB Primer*, Dept. of Math., Univ. of Fla. (3d ed. 1993) ("Matlab").

The Examiner rejected claims 2-10, 55-63, and 114-117 under 35 U.S.C. § 102(b) as anticipated by Matlab (Ans. 4-15).

Rather than repeat the arguments of Appellants or the Examiner, we refer to the Briefs and the Answer[3] for their respective details. In this decision, we have considered only those arguments actually made by Appellants. Arguments which Appellants could have made but did not make in the Briefs have not been considered and are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Claims 5-10 and 58-63

Regarding independent claims 5 and 58,[4] Appellants argue that Matlab does not restore a previous version of a three-dimensional mesh model on a computer system as claimed. Specifically, Appellants contend that Matlab does not describe an ordered list of computer operations containing the operations which, if performed in order on an earlier state of

[3] Throughout this opinion, we refer to (1) the Appeal Brief filed Dec. 19, 2007; (2) the latest Examiner's Answer mailed Feb. 28, 2008; and (3) the Reply Brief filed Apr. 28, 2008.

[4] Appellants argue claims 5-10 and 58-63 together as a group. *See* App. Br. 4-6. Accordingly, we treat these claims separately from claims 2-4, 55-57, and 114-117, which were separately argued (App. Br. 6-8). *See* 37 C.F.R. § 41.37(c)(1)(vii).

the mesh model, would result in a current state of the model as claimed. Although Appellants concede that a user could *possibly* create an "M-file" in Matlab to implement the recited functionality, Appellants emphasize that Matlab simply does not indicate that a user has actually done so. As such, Appellants contend, Matlab does not *necessarily* perform the recited method and therefore does not anticipate the claimed invention. (App. Br. 5-6; Reply Br. 2-3; emphases added.)

The Examiner, however, contends that the claimed method is inherently performed in Matlab's normal use and operation. In reaching this conclusion, the Examiner reasons that since Matlab motivates and encourages the user to explore and experiment with the various capabilities of the system, the claimed process would have been implemented in the course of normal operation of Matlab. (Ans. 18-23.)

Claims 2-4, 55-57, and 114-117

Appellants make a similar argument with respect to the limitations of independent claims 115-117. According to Appellants, the mere fact that a user could hypothetically have carried out or assembled the claimed invention using Matlab's programming functionality does not anticipate the claims. (App. Br. 7-8.)

The Examiner essentially reiterates the previously-noted position regarding Matlab, but adds that Matlab's M-file functionality fully meets the recited three states of the mesh model, particularly since M-files can reference themselves recursively (Ans. 23-29). The Examiner further notes that system claim 117 merely recites the intended function of the computer

modules, but does not recite that they actually do anything, let alone the disputed process steps (Ans. 24).

The issues before us, then, are as follows:

ISSUES

(1) Have Appellants shown that the Examiner erred in rejecting independent claims 5 and 58 under § 102 by finding that Matlab necessarily restores a previous version of a three-dimensional mesh model on a computer system by performing at least some operations from an ordered list of operations which, if performed in order on an earlier state of the mesh model, would result in a current state of the model as claimed?

(2) Have Appellants shown that the Examiner erred in rejecting independent claims 115-116 under § 102 by finding that Matlab necessarily manages a three-dimensional mesh model on a computer system by (1) storing a copy of a first state of the model on the computer system; (2) performing operations on the model where the model is in a second state after the operations; (3) storing a record of each of the operations in an ordered list on the computer system; and (4) reapplying at least some of the operations in the list to the stored first state of the model to arrive at a third state after reapplying the operations?

(3) Have the Appellants shown that the Examiner erred in finding that the computer modules recited in system claim 117 and dependent claim 114 are capable of performing the recited functions?

FINDINGS OF FACT

The record supports the following findings of fact (FF) by a preponderance of the evidence:

1. Matlab describes an interactive, matrix-based system (MATLAB)[5] for scientific and engineering numeric computations and visualization (Matlab, at ii).

2. Matlab enables the user to enter matrices via a file by using the "load" command or via a script file (Matlab, at 1-2).

3. Matlab can execute a sequence of statements in files known as "M-files." One type of M-file is a script file that contains a sequence of normal Matlab statements. (Matlab, at 9.)

4. "An M-file can reference other M-files, including referencing itself recursively." (Matlab, at 9.)

5. Users can save a Matlab session to a file which enables restoring the workspace to its former state when reentering the system (i.e., via the "load" command) (Matlab, at 3 ("Saving a session")).

6. The "mesh" command draws three-dimensional wire mesh surface plots, and the command "mesh (z)" creates a three-dimensional perspective plot of the elements of matrix "z" (Matlab, at 18).

7. Matlab has a command line editing and recall feature that enables the user to scroll through a stack of previous commands using the up/down arrows. With this functionality, the user can (1) recall a previous command line; (2) edit it; and (3) execute the revised command line. For example,

---

[5] "The name MATLAB is derived from MATrix LABoratory." (Matlab, at ii.)

"flopcounts"[6] for computing the inverse of matrices of various sizes could be compared by repeatedly recalling, editing, and executing. (Matlab, at 8.)

8. Matlab repeatedly encourages the user to try various features of the system. *See, e.g.*, Matlab, at 7 (encouraging user to "[t]ry it" in connection with vector function example), 8 (same for matrix function example), 9 (same for generating a table of sines), 14 (same for comparing efficiency of algorithms), 15 (same for drawing graph of sine function); 18 (noting that a user can draw a graph of a function using the "mesh (z)" command).

PRINCIPLES OF LAW

Anticipation is established only when a single prior art reference discloses, expressly or under the principles of inherency, each and every element of a claimed invention as well as disclosing structure which is capable of performing the recited functional limitations. *RCA Corp. v. Appl. Dig. Data Sys., Inc.*, 730 F.2d 1440, 1444 (Fed. Cir. 1984); *W.L. Gore & Assoc., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1554 (Fed. Cir. 1983).

"Inherency . . . may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999) (citations omitted).

To avoid anticipation, apparatus claims must be distinguished from the prior art in terms of structure rather than function. *In re Schreiber*, 128 F.3d 1473, 1477-78 (Fed. Cir. 1997). Thus, even if a prior art device is used for a different purpose, it will nevertheless anticipate the claim if it expressly

---

[6] A flopcount pertains to the number of floating point operations (flops) performed and is indicative of an algorithm's efficiency. *See* Matlab, at 14.

or inherently contains all claimed structural features and is capable of performing the intended function. *See id.*

ANALYSIS

Claims 5-10 and 58-63

This appeal hinges on one crucial question: whether providing the *capability* to implement a recited function in a known computer system necessarily *performs* that function. Indeed, Appellants all but admit that Matlab is capable of being programmed to implement the functionality of claim 5 if a user were so inclined. *See* App. Br. 5 (noting that it may be possible for a user to create an M-file in Matlab that implements the recited functionality of claim 5). Clearly, this is a possibility. But since anticipation cannot be based on possibilities or probabilities, but rather must be based on what is *necessarily present* in the reference either expressly or inherently, *Robertson*, 169 F.3d at 745, we must therefore evaluate the Examiner's anticipation rejection in light of this mandate.

As an initial matter, we note that the scope of the claim language does not preclude the "earlier state" of the model to match that of the "current state" of the model. Claim 5 merely requires performing *at least some* of the operations in the ordered list on the retrieved copy of the model. Performing "at least some" of the operations does not preclude performing *all* of the operations. As such, this step is fully met by merely repeating the same operations that were used to obtain the earlier state of the model (i.e., in an earlier session) in arriving at a current state of the model in a subsequent session. Put another way, the limitation would be fully met by automatically

restoring a saved version of a model (i.e., a copy of the "earlier state" of the model) on a computer system in a subsequent session.

With these considerations in mind, we turn to Matlab. As the Examiner indicates (Ans. 4), Matlab enables users to restore a workspace to its former state when they reenter the system (FF 5). While Matlab does not expressly indicate that this restoration necessarily occurs in connection with a three-dimensional mesh model as claimed, Matlab nonetheless provides a three-dimensional mesh modeling capability (FF 6)—a finding that is undisputed.

The obvious inference from these respective features is that users can restore workspaces involving any of a wide variety of functions of Matlab, including three-dimensional mesh modeling. But while users *can* restore workspaces with such models, they need not do so. As such, we fail to see how the specifically-recited functionality of claim 5 is *necessarily* present in Matlab.

We reach the same conclusion with respect to Matlab's M-file capability (FF 3-5)—a capability that Appellants acknowledge could provide the functionality of the claimed invention if such files were so programmed (App. Br. 5). Here again, for anticipation, the question is not whether the functionality *could* be obtained via this feature of Matlab, but rather whether *it actually is* obtained such that it is *necessarily* present in the reference. On the record before us, we simply cannot say that this is the case.

Furthermore, we find unavailing the Examiner's reliance on Matlab's repeated encouraging users to explore and try various system features (Ans. 19-22). While Matlab does repeatedly urge and encourage users to try various system features (FF 8), the Examiner has not identified—nor can we

find—anything in Matlab that specifically urges users to try the particular combination of features in Matlab that are said to anticipate claim 5. At best, we would have to infer that, based on this "encouragement," the user would necessarily have tried to restore a workspace involving a three-dimensional mesh model or programmed an M-file to achieve the functionality of claim 5. We simply cannot say that this is the case. While the Examiner's logic may have some relevance to an obvious-to-try analysis under § 103, it has no place in an anticipation rejection.

As Appellants indicate (Reply Br. 3), to hold that the mere existence of the commands of a programming language (e.g., Matlab's M-files) anticipates a particular process that is achievable via a particular application of those commands would render virtually any computer-implemented process inherently anticipated by the commands themselves—even if the computer had never been so programmed. Such a result is hardly consistent with our precedents regarding anticipation, and Appellants' cogent analysis in this regard (Reply Br. 3) is precisely why the Examiner's position is untenable.

The Examiner's reliance (Ans. 21) on *Perricone v. Medicis Pharm. Corp.*, 432 F.3d 1368 (Fed Cir. 2005) only bolsters our conclusion. In that case, the issue was whether a prior art reference (Pereira) that disclosed a topical application of a lotion inherently anticipated applying the disclosed composition to a *skin sunburn*. Or, as the court put it, "[t]he issue is not . . . whether Pereira's lotion, *if applied* to skin sunburn would inherently treat that damage, but whether Pereira discloses the application of its composition to skin sunburn. It does not." *Id.* at 1378 (emphasis in original).

This distinction is strikingly similar to the issue before us in the present appeal. That is, the issue before us is not whether Matlab's system *if suitably programmed or used* would inherently anticipate the process of claim 5, but whether the reference actually discloses such a process. And like the prior art reference in *Perricone*, Matlab falls short in this regard.

We acknowledge that the MPEP indicates that a prior art device anticipates a claimed method when the disclosed device, in its normal and usual operation, *necessarily* performs the method claimed. MPEP § 2112.02, Rev. 6, Sept. 2007 ("MPEP") (emphasis added). We emphasize the term "necessarily" since it is a critical qualifier in this instruction as Appellants indicate (Reply Br. 2).

While it is undisputed that Matlab *could* perform the method of claim 5 in its normal and usual operation if it were suitably programmed, Matlab does not *necessarily* perform the claimed method in its normal and usual operation—at least given the specific functions described in the reference. Rather, to achieve the recited functionality of claim 5 using Matlab, the user must introduce extra undisclosed functionality (e.g., via programming), or at least relate certain functions to each other in a particular way that goes beyond that actually described in the reference. In short, *but for* this extra functionality imparted by the user, the functionality of claim 5 would not occur—even during the "normal and usual operation" of the device. Therefore, the functionality of claim 5 does not *necessarily* occur during system's normal and usual operation. As such, Matlab fails to anticipate claim 5.

For the foregoing reasons, Appellants have persuaded us of error in the Examiner's rejection of independent claim 5 and independent claim 58

which recites commensurate limitations. Therefore, we will not sustain the Examiner's rejection of those claims, and dependent claims 6-10 and 59-63 for similar reasons.

Claims 2-4, 55-57, 115, and 116

We reach a similar conclusion with respect to independent claims 115 and 116 which call for, in pertinent part, managing a three-dimensional mesh model on a computer system by (1) storing a copy of a first state of the model on the computer system; (2) performing operations on the model where the model is in a second state after the operations; (3) storing a record of each of the operations in an ordered list on the computer system; and (4) reapplying at least some of the operations in the list to the stored first state of the model to arrive at a third state after reapplying the operations.

In reaching this conclusion, however, we note that the Examiner's point (Ans. 23) regarding the M-file functionality as corresponding to a particular state (i.e., the recited first, second, and third state) is well-taken. We add that Matlab compares different "states" of a particular function by repeatedly recalling, editing, and executing previously-executed command lines (FF 7). Nevertheless, we still fail to see how the specific recited functionality of claims 115 and 116 would *necessarily* occur during the normal and usual operation of Matlab essentially for the reasons indicated in connection with claims 5 and 58. In short, there is nothing in Matlab that expressly or inherently indicates that this particular functionality would *necessarily* be used in connection with a three-dimensional mesh model in the manner recited in claims 115 and 116. We therefore cannot sustain the

Examiner's anticipation rejection of those claims and dependent claims 2-4 and 55-57 for similar reasons.

Claims 114 and 117

We reach the opposite conclusion, however, with respect to claims 114 and 117. As the Examiner indicates (Ans. 24), representative independent claim 117 recites a system with various computer modules which, while reciting functionality commensurate with the process of claim 115, merely do so as an *intended use* of the modules. But these modules need not actually perform this intended use to meet the claim: rather, the modules need only be *capable* of such intended use. *See Schreiber*, 128 F.3d at 1477-78 (emphasis added).

On the record before us, we see no reason why Matlab's modules would not be so capable. First, as we noted previously, Appellants all but admit that Matlab is capable of being programmed to implement the functionality of claim 5. *See* App. Br. 5 (noting that it may be possible for a user to create an M-file in Matlab that implements the recited functionality of claim 5). We see no reason why Matlab could not likewise be capable of the functionality of claim 117, particularly in view of the Examiner's reliance on Matlab as corresponding to the recited states of the model—a position that we find reasonable given that M-files can reference themselves recursively (FF 4) as the Examiner indicates (Ans. 23).

We therefore find that Matlab is capable of performing the recited functions of the computer modules of claim 117. In reaching this conclusion, we emphasize that, unlike method claim 115, claim 117 does not

positively recite performing the recited steps. The Examiner emphasizes this distinction and, in doing so, actually illustrates how the claim could be modified to require performing the steps. *See* Ans. 24 (suggesting alternative claim language for claim 117 to perform the recited functions). The Examiner's point in this regard is well taken.

For the foregoing reasons, Appellants have not persuaded us of error in the Examiner's anticipation rejection of representative claim 117. Therefore, we will sustain the Examiner's rejection of that claim, and claim 114 which falls with claim 117.

CONCLUSIONS

(1) Appellants have shown that the Examiner erred in rejecting independent claims 5 and 58 under § 102 by finding that Matlab necessarily restores a previous version of a three-dimensional mesh model on a computer system by performing at least some operations from an ordered list of operations which, if performed in order on an earlier state of the mesh model, would result in a current state of the model as claimed. Furthermore, claims 6-10 and 59-63 stand with claims 5 and 58 from which they depend.

(2) Appellants have shown that the Examiner erred in rejecting independent claims 115-116 under § 102 by finding that Matlab necessarily manages a three-dimensional mesh model on a computer system by (1) storing a copy of a first state of the model on the computer system; (2) performing operations on the model where the model is in a second state after the operations; (3) storing a record of each of the operations in an ordered list on the computer system; and (4) reapplying at least some of the operations in the list to the stored first state of the model to arrive at a third

state after reapplying the operations. For similar reasons, claims 2-4 and 55-57 stand with claims 115 and 116 from which they depend.

(3) Appellants have not shown that the Examiner erred in finding that the computer modules recited in system claim 117 and dependent claim 114 are capable of performing the recited functions.

Thus, Appellants have shown that the Examiner erred in rejecting claims 2-10, 55-63, 115, and 116 under § 102. Appellants, however, have not shown that the Examiner erred in rejecting claims 114 and 117 under § 102.

ORDER

The Examiner's decision rejecting claims 2-10, 55-63, and 114-117 is affirmed-in-part.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED-IN-PART

pgc

KENYON & KENYON LLP
ONE BROADWAY
NEW YORK, NY 10004